Soluții REGEX pentru notepad++ mai puțin cunoscute, dar GENIALE

Selecteaza fisierul care nu contine <!-- SASA-2 -->

Solutie: (?!.*<!-- SASA-2 -->)

aceasta cautare verifica daca <!-- SASA-2 --> nu se gaseste in fisier

Explicația componentelor:

(?s) – Activează modul "dot-all", permițând punctului (.) să includă și caracterele de rând nou (\n). Este esențial pentru a scana tot fișierul ca pe un singur bloc de text.

\A – Ancora care marchează începutul absolut al fișierului.

(?!.*<!-- SASA-2 -->) – Un negative lookahead care verifică dacă, oriunde în fișier, există secvența specificată. Dacă o găsește, potrivirea eșuează imediat.

.* – Consumă restul fișierului pentru a confirma potrivirea dacă lookahead-ul a avut succes.

Stack Overflow

Stack Overflow

+6

Unde îl poți folosi:

Notepad++: În fereastra "Find in Files", bifează modul Regular expression și asigură-te că opțiunea . matches newline este activă (sau folosește prefixul (?s)).

VS Code: Poți folosi un lookahead similar în câmpul de căutare (Search), activând iconița .* (Use Regular Expression).

Linia de comandă (grep): Dacă preferi terminalul, este mai eficient să folosești parametrul -L (files-without-match) în loc de un regex complex:

bash

grep -L "<!-- SASA-2 -->" *.*

Folosește codul cu precauție.

Această comandă va lista direct numele fișierelor care nu conțin textul

---

acuma vreau sa gasesc ca <!-- SASA-2 --> nu exista intre si <!-- ARTICOL FINAL --> din ecuatia

* Note: Flawless (2007)

<!----------POP-UP---------->

<!-- ARTICOL FINAL -->

SOLUTIE: (?s) (?!(?:(?!<!-- ARTICOL FINAL -->).)*<!-- SASA-2 -->).*?<!-- ARTICOL FINAL -->

Cum funcționează „magia”:

(?s): Permite punctului (.) să includă și liniile noi (multiline).

: Găsește punctul de început al căutării.

(?!(?:(?!<!-- ARTICOL FINAL -->).)*<!-- SASA-2 -->): Acesta este un Negative Lookahead. El spune: "Uită-te înainte și asigură-te că NU găsești <!-- SASA-2 --> înainte de a ajunge la <!-- ARTICOL FINAL -->".

(?!<!-- ARTICOL FINAL -->). : Această parte se asigură că scanarea nu trece de finalul secțiunii tale.

.*?<!-- ARTICOL FINAL -->: Dacă verificarea de la punctul 3 a reușit (deci textul căutat lipsește), atunci selectează tot conținutul până la eticheta de final.

De ce este mai bun acest model?

Dacă ai mai multe astfel de blocuri în același fișier, această formulă le va identifica doar pe cele "defecte" (cele cărora le lipsește eticheta SASA-2). Formula ta inițială cu \A ar fi verificat întregul fișier ca pe un tot unitar, pe când aceasta lucrează pe segmente.

Sfaturi de utilizare în Notepad++ / VS Code:

Asigură-te că ai bifat Regular expression.

Dacă vrei să înlocuiești sau să adaugi automat codul acolo unde lipsește, poți folosi grupuri de captură, dar este recomandat să faci o căutare simplă mai întâi pentru a valida rezultatele.

Ai dori să te ajut să construiesc și formula de înlocuire (Replace) pentru a insera automat <!-- SASA-2 --> acolo unde lipsește?

---

da, si daca vreau sa imi arate ca nu se gaseste <!-- SASA-2 --> doar intre ultima instanta si care mai apoi e urmat de <!-- ARTICOL FINAL -->

Deci, selecteaza fisierul CARE NU CONTINE <!-- SASA-2 --> din perimetrul urmator:

* Note: Flawless (2007)

<!----------POP-UP---------->

<!-- ARTICOL FINAL -->

REZOLVARE: Hai să folosim o variantă simplificată, care nu mai dă erori de sintaxă în Notepad++ și care găsește exact ce ai tu în imagine:

Formula de Căutare (Find)

(?s) (?:(?!<!-- SASA-2 -->|).)*? .*?<!-- ARTICOL FINAL -->

DAca vreai sa adaugi <!-- SASA-2 --> intre si din acelasi perimetru:

FIND:

(?s) ((?:(?!<!-- SASA-2 -->|).)*?)(.*?<!-- ARTICOL FINAL -->)

Replace with:

\1\n   <!-- SASA-2 -->\n\2

Folosește codul cu precauție.

Ce face această înlocuire:

: Menține tag-ul de început.

\1: Pune înapoi tot textul/spațiile care erau între și .

\n <!-- SASA-2 -->\n: Inserează eticheta ta pe un rând nou, frumos aliniat.

\2: Pune la loc restul blocului (, POP-UP și ARTICOL FINAL).

Sfat: Rulează Replace pe un singur exemplu (Find Next -> Replace) ca să verifici dacă alinierea este așa cum îți place înainte de a da Replace All.

---

acuma vreau sa gasesc ca <!-- SASA-2 --> nu exista intre si <!-- ARTICOL FINAL --> din ecuatia

* Note: Flawless (2007)

 <!----------POP-UP---------->

Formula gasita mai devreme este buna, dar trebuie sa apas de doua ori butonul Replace sau Replace All ca sa faca inlocuirea.

FIND: (?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)

Replace: </p>\1\n   <!-- SASA-2 -->\n\2

acuma am formula asta care merge din prima, de la primul replace:

FIND:

(?-si:</p>|(?!\A)\G)(?s-i:(?!<\!-- ARTICOL FINAL -->).)*?\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)

Replace:

</p>\1\n   <!-- SASA-2 -->\n\2

Dar uite cum am ajuns la aceasta formula:

VOM FOLOSI GENERIC FORMULAS:

GENERIC FORMULAS

Lista Generic Regex:

(?:.*?BSR|\G).*?FR\K(.*?)(?=ESR.*?ESR)

(?-si:BSR|(?!\A)\G)(?s-i:(?!ESR).)*?\KFR

(BSR)\s*FR(.*?)FR\s*(ESR) Replace by: \1\2\3

(?s-i)(?:BSR|(?!\A)\G)(?:(?!ESR).)*?\K(FR)

(BSR.*?)(FR\s*ESR)

(?:BSR|\G).*?\K(?:FR(*SKIP)(*FAIL)|ESR)

(?:BSR|\G).*?\K(?:FR(*SKIP)(*FAIL)|<(?:(?!ESR).)+?>)

(?:BSR|\G).*?\K(?:FR(*SKIP)(*FAIL)|<(?:(?!ESR1|ESR2).)+?>)

BSR(?:[^<]+|<(?!FR1|FR2))?</p>(*SKIP)(FAIL)|<(?!FR1|FR2)[^>]+>

BSR(?:[^<]+|<(?!FR))?</p>(*SKIP)(FAIL)|<(?!FR)[^>]+>

(?-s)(?-i:BSR|(?!\A)\G)(?s-i:.*?ESR)(*SKIP)(*FAIL)|FR

(?s)(?-i:BSR|(?!\A)\G).*?\K(?!ESR)

(?-si:BSR|(?!\\A)\\G)(?s-i:(?!ESR).)*?\\x20\\K(FR)

(?-si:BSR|(?!\\A)\\G)(?s-i:(?!ESR).)*?\\x20\\KFR(?=\\x20)

(?-si:BSR|(?!\\A)\\G)(?s-i:(?!ESR).)*?\\x20\\KFR

(?-si:BSR|(?!\\A)\\G)(?s-i:(?!ESR).)*?\\x20\\KFR(?=\\x20)

(?s-i)(?:BSR|(?!\A)\G)(?:(?!ESR).)*?\K<(?!(?:FR1|FR2))[^>]*>

(?s)(?-i:BSR|(?!\A)\G).*?\K(?<=FR1)(.*?)(?=FR2)(?!ESR)

(?s)(?-i:BSR).*?(?<=FR1)(.*?)(?=FR2)(?=.*?ESR)

BSR[\s\S]+FR1\s+(\S+)\s+FR2[\s\S]+ESR

(?s)(?<=BSR[\s\S]+FR1\s+)(\S+)(?=\s+FR2[\s\S]+ESR)

(?s)BSR[\s\S]+FR1\s+(\S+)\s+FR2[\s\S]+ESR

(?-s)(?-i:BSR|(?!\A)\G)(?s-i:.*?ESR)(*SKIP)(*FAIL)|FR

(?-s)(?-i:BSR|(?!\A)\G).*?\KFR(?!ESR)

(?-s)(?-i:BSR|(?!\A)\G)(?s-i:(?!ESR).)*?\K(?-si:FR)

(?-s)(?-i:BSR|(?!\A)\G).*?\K(?-si:FR)(?=(?s-i:.*?ESR))

(?s)BSR(FR)(?-s).+ESR or BSR(FR).+(?s:(?=.*(ESR)))

(?s)(?-i:BSR|(?!\A)\G).*?\KFR(?!ESR)

(?s)(?-i:BSR|(?!\A)\G)(?s-i:(?!ESR).)*?\K(?-si:FR)

(?s)(?-i:BSR|(?!\A)\G).*?\K(?-si:FR)(?=(?s-i:.*?ESR))

(?s)BSR(FR)(?-s).+ESR or BSR(FR).+(?s:(?=.*(ESR)))

(?-si:BSR|(?!\A)\G).*?\KFR(?!ESR)

(?-si:BSR|(?!\A)\G)(?s-i:(?!ESR).)*?\K(?-si:FR)

(?-si:BSR|(?!\A)\G).*?\K(?-si:FR)(?=(?s-i:.*?ESR))

(?s)BSR(FR)(?-s).+ESR sau BSR(FR).+(?s:(?=.*(ESR)))

(?-s)(?-i:BSR|(?!\A)\G).*?\K(?-i:FR)(?=ESR)

(?-si:BSR|(?!\A)\G)(?s-i:(?!ESR).)*?\K(FR)+ REPLACE BY: \2 sau \3

(?-i:BSR|\G(?!^))(?s:(?!ESR).)*?\K(?-i:FR)

BSR(?!\A)(?-s:.*?)\K(?-si:FR)(?=(?s-i:.*?ESR))

BSR\K.*?(?!\A)(?-s:.*?)\K(?-si:FR)(?=(?s-i:.*?ESR))

(?-i:BSR|(?!\A)\G)(?s:(?!ESR).)*?\K(?-i:FR)

(?-i:BSR|(?!^)\G)(?s:(?!ESR).)*?\K(?-i:FR)

(?s)\A.*BSR(FR)*ESR.*\z|\A.*BSR(FR)+ESR.*\z

(?s)\A.*BSR(FR)+ESR.*\z

(?s)\A.*BSR+(FR)+ESR.*\z

(?:BSR|\G)\S*\K(?:FR)(?=.*ESR)

(?:BSR|\G)\S*\K(FR)(?=.*ESR)

(?:BSR|\G)\S*\K(?:\h+FR)(?=.*ESR)

(?<!=\x20)(?<!=)FR(?!>|ESR|\x20>|\x20/>|\?>|\x20\?>|BSR)

(?<!=\x20)(?<!=)FR(?!>|ESR|\x20>|\x20/>|\?>|\x20\?>|\x20BSR)

(?<!=\x20)(?<!=)FR(?!>|ESR|\x20\?>|\x20BSR)

(?-is)^(?!.*BSR)(?!.*ESR)(?:(FR))

(?-is)^(?!.*BSR)(?!.*ESR)(?:(FR)(.+))

(BSR).*\K(?:\h+(FR))(?=.*ESR)

(?-i:BSR|\G(?!^))(?s:(ESR).)*?\K(?-i:FR)

(BSR)+(.)+\K(FR)(?s:(?=.*(ESR)))

(BSR)(FR)(.*?)(ESR)

(?s-i)^.+BSR\R|ESR.+|(?-s)^(FR)|(.+))$

(?s-i)^.+BSR\R|ESR.+|(?-s)^(FR)(.+))$

(BSR)(.*?)(FR)(.*?)(ESR)

(?s)(?-i:(BSR).+?ESR|\G(?!^))((?!(ESR)).)*?\K(FR)

(?s)(?-i:BSR.+?ESR|\G(?!^))((?!ESR).)*?\KFR

(BSR)+(.)+\K(FR)(?=.*ESR)

(BSR)+(.)+\K(FR)(?=.*)(ESR)

(BSR)*\K(?:(FR))(?=.*ESR)

(BSR)\K(?-s)(.*?)\KFR(?=.*ESR)

.*?(BSR)(?-s)(.*?)(FR).*(.*ESR)

(BSR).*(\b(FR)\b.*).*(ESR)

(?:BSR|\G).*?\K(?:FR(*SKIP)(*FAIL)|<(?!ESR)[^>]*>)

BSR(?=(?:(?!ESR).)*?FR).+?ESR\R?

BSR(?=(?:(?!FR).)*?FR).+?ESR\R?

BSR(?=(?:(?!ESR).)*?FR)(?=(?:(?!ESR).)*?FR).+?ESR\R?

BSR(?=(?:(?!ESR).)*?FR)(?=(?:(?!ESR).)).+?ESR\R?

```
(?s)(?:BSR|\G)(?:(?!ESR).)*?\K(FR)
```

```
(?s)(?:BSR|\G(?!^))(?:(?!ESR).)*?\KFR
```

```
(?!BSR)*\K(?!BSR)(?:(FR)))(?=.*ESR)
```

```
(?s)(?:BSR|\G)(?:(?!ESR).)*?\K(?:FR)
```

```
(?s)(?:BSR|(?!\A)\G)(?:(?!ESR).)*?\KFR
```

```
(?s)(?:BSR|\G).*?\K(?:FR(*SKIP)(*FAIL)|<(?:(?!ESR).)+?>)
```

```
(?:BSR|\G).*?\K(?:FR(*SKIP)(*FAIL)|<(?:(?!ESR).)+?>)
```

```
(?:BSR|\G).*?\K).*?(FR)(?=.*ESR)
```

```
(?:BSR|\G).*?\K)(FR).*?(?=.*ESR)
```

```
(?:BSR|\G)(FR).*?\K).*?(?=.*ESR)
```

```
(?s-i)(?:BSR|\G)(?:(?!ESR).)*?\K(FR)
```

```
(?:BSR|\G)(FR).*?(FR)\K).*?(?=.*ESR)
```

```
(?s-i)(?:<p[^>]*>|(?!\A)\G)(?:(?!</p>).)*?\K<(?!(?:/[ap]>|a\x20[^>]+>))[^>]*>
```

```
(?s-i)(?:BSR|(?!\A)\G)(?:(?!ESR).)*?\K<(?!(?:FR1|FR2))[^>]*>
```

(?:BSR|\G).*?\K(?:FR(*SKIP)(*FAIL)|<(?:(?!ESR).)+?>)

(?:BSR|\G).*?\K(?:FR(*SKIP)(*FAIL)|<(?:(?!ESR1|ESR2).)+?>)

BSR(?:[^<]+|<(?!FR1|FR2))?ESR(*SKIP)(FAIL)|<(?!FR1|FR2)[^>]+>

SPECIFICATII:

Let BSR (Begin Search-region Regex) be the regex which defines the beginning of the area where the search for FR, must start

Let ESR (End Search-region Regex) be the regex which defines, implicitly, the area where the search for FR, must end

Let FR (Find Regex) be the regex which defines the char, string or expression to be searched

Let RR (Replacement Regex) be the regex which defines the char, string or expression which must replace the FR expression. REPLACE RR (means can be \1 \2 ...or $1 $2 or other

Deci, in cazul nostru, vom inlocui astfel:

BSR = </p>

ESR = <\!-- ARTICOL FINAL -->

FR = (?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->) aceasta cautare verifica daca <!-- SASA-2 --> nu se gaseste in fisier

sau alternativ:

FR = (?s)\A(?!.*<!-- SASA-2 -->).* aceasta cautare verifica daca <!-- SASA-2 --> nu se gaseste in fisier

DUPA INLOCUIRE:

Lista Generic Regex:

(?:.*?</p>|\G).*?(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)\K(.*?)(?=<\!-- ARTICOL FINAL -->.*?<\!-- ARTICOL FINAL -->)

(?-si:</p>|(?!\A)\G)(?s-i:(?!<\!-- ARTICOL FINAL -->).)*?\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->) - asta e buna

(</p>)\s*(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(.*?)(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)\s*(<\!-- ARTICOL FINAL -->) Replace by: \1\2\3

(?s-i)(?:</p>|(?!\A)\G)(?:(?!<\!-- ARTICOL FINAL -->).)*?\K((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(</p>.*?)((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)\s*<\!-- ARTICOL FINAL -->)

(?:</p>|\G).*?\K(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(*SKIP)(*FAIL)|<\!-- ARTICOL FINAL -->)

(?:</p>|\G).*?\K(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(*SKIP)(*FAIL)|<(?:(?!<\!-- ARTICOL FINAL -->).)+?>)

(?:</p>|\G).*?\K(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(*SKIP)(*FAIL)|<(?:(?!<\!-- ARTICOL FINAL -->1|<\!-- ARTICOL FINAL -->2).)+?>)

</p>(?:[^<]+|<(?!(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1|(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2))?</p>(*SKIP)(FAIL)|<(?!(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1|(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2)[^>]+>

</p>(?:[^<]+|<(?!(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)))?</p>(*SKIP)(FAIL)|<(?!(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))[^>]+>

(?-s)(?-i:</p>|(?!\A)\G)(?s-i:.*?<\!-- ARTICOL FINAL -->)(*SKIP)(*FAIL)|(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)

(?s)(?-i:</p>|(?!\A)\G).*?\K(?!<\!-- ARTICOL FINAL -->)

(?-si:</p>|(?!\\A)\\G)(?s-i:(?!<\!-- ARTICOL FINAL -->).)*?\\x20\\K((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(?-si:</p>|(?!\\A)\\G)(?s-i:(?!<\!-- ARTICOL FINAL -->).)*?\\x20\\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(?=\\x20)

(?-si:</p>|(?!\\A)\\G)(?s-i:(?!<\!-- ARTICOL FINAL -->).)*?\\x20\\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)

(?-si:</p>|(?!\\A)\\G)(?s-i:(?!<\!-- ARTICOL FINAL -->).)*?\\x20\\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(?=\\x20)

(?s-i)(?:</p>|(?!\A)\G)(?:(?!<\!-- ARTICOL FINAL -->).)*?\K<(?!(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1|(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2))[^>]*>

(?s)(?-i:</p>|(?!\A)\G).*?\K(?<=(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1)(.*?)(?=(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2)(?!<\!-- ARTICOL FINAL -->)

(?s)(?-i:</p>).*?(?<=(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1)(.*?)(?=(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2)(?=.*?<\!-- ARTICOL FINAL -->)

</p>[\s\S]+(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1\s+(\S+)\s+(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2[\s\S]+<\!-- ARTICOL FINAL -->

(?s)(?<=</p>[\s\S]+(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1\s+)(\S+)(?=\s+(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2[\s\S]+<\!-- ARTICOL FINAL -->)

(?s)</p>[\s\S]+(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1\s+(\S+)\s+(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2[\s\S]+<\!-- ARTICOL FINAL -->

(?-s)(?-i:</p>|(?!\A)\G)(?s-i:.*?<\!-- ARTICOL FINAL -->)(*SKIP)(*FAIL)|(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)

(?-s)(?-i:</p>|(?!\A)\G).*?\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(?!<\!-- ARTICOL FINAL -->)

(?-s)(?-i:</p>|(?!\A)\G)(?s-i:(?!<\!-- ARTICOL FINAL -->).)*?\K(?-si:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(?-s)(?-i:</p>|(?!\A)\G).*?\K(?-si:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=(?s-i:.*?<\!-- ARTICOL FINAL -->))

(?s)</p>((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?-s).+<\!-- ARTICOL FINAL --> or </p>((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).+(?s:(?=.*(<\!-- ARTICOL FINAL -->)))

(?s)(?-i:</p>|(?!\A)\G).*?\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(?!<\!-- ARTICOL FINAL -->)

(?s)(?-i:</p>|(?!\A)\G)(?s-i:(?!<\!-- ARTICOL FINAL -->).)*?\K(?-si:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(?s)(?-i:</p>|(?!\A)\G).*?\K(?-si:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=(?s-i:.*?<\!-- ARTICOL FINAL -->))

(?s)</p>((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?-s).+<\!-- ARTICOL FINAL --> or </p>((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).+(?s:(?=.*(<\!-- ARTICOL FINAL -->)))

(?-si:</p>|(?!\A)\G).*?\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(?!<\!-- ARTICOL FINAL -->)

(?-si:</p>|(?!\A)\G)(?s-i:(?!<\!-- ARTICOL FINAL -->).)*?\K(?-si:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(?-si:</p>|(?!\A)\G).*?\K(?-si:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=(?s-i:.*?<\!-- ARTICOL FINAL -->))

(?s)</p>((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?-s).+<\!-- ARTICOL FINAL --> sau </p>((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).+(?s:(?=.*(<\!-- ARTICOL FINAL -->)))

(?-s)(?-i:</p>|(?!\A)\G).*?\K(?-i:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=<\!-- ARTICOL FINAL -->)

(?-si:</p>|(?!\A)\G)(?s-i:(?!<\!-- ARTICOL FINAL -->).)*?\K((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))+ REPLACE BY: \2 sau \3

(?-i:</p>|\G(?!^))(?s:(?!<\!-- ARTICOL FINAL -->).)*?\K(?-i:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

</p>(?!\A)(?-s:.*?)\K(?-si:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=(?s-i:.*?<\!-- ARTICOL FINAL -->))

</p>\K.*?(?!\A)(?-s:.*?)\K(?-si:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=(?s-i:.*?<\!-- ARTICOL FINAL -->))

(?-i:</p>|(?!\A)\G)(?s:(?!<\!-- ARTICOL FINAL -->).)*?\K(?-i:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(?-i:</p>|(?!^)\G)(?s:(?!<\!-- ARTICOL FINAL -->).)*?\K(?-i:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(?s)\A.*</p>((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))*<\!-- ARTICOL FINAL -->.*\z|\A.*</p>((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))+<\!-- ARTICOL FINAL -->.*\z

(?s)\A.*</p>((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))+<\!-- ARTICOL FINAL -->.*\z

(?s)\A.*</p>+((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))+<\!-- ARTICOL FINAL -->.*\z

(?:</p>|\G)\S*\K(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=.*<\!-- ARTICOL FINAL -->)

(?:</p>|\G)\S*\K((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=.*<\!-- ARTICOL FINAL -->)

(?:</p>|\G)\S*\K(?:\h+(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=.*<\!-- ARTICOL FINAL -->)

(?<!=\x20)(?<!=)(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(?!>|<\!-- ARTICOL FINAL -->|\x20>|\x20/>|\?>|\x20\?>|</p>)

(?<!=\x20)(?<!=)(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(?!>|<\!-- ARTICOL FINAL -->|\x20>|\x20/>|\?>|\x20\?>|\x20</p>)

(?<!=\x20)(?<!=)(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(?!>|<\!-- ARTICOL FINAL -->|\x20\?>|\x20</p>)

(?-is)^(?!.*</p>)(?!.*<\!-- ARTICOL FINAL -->)(?:((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)))

(?-is)^(?!.*</p>)(?!.*<\!-- ARTICOL FINAL -->)(?:((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(.+))

(</p>).*\K(?:\h+((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)))(?=.*<\!-- ARTICOL FINAL -->)

(?-i:</p>|\G(?!^))(?s:(<\!-- ARTICOL FINAL -->).)*?\K(?-i:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(</p>)+(.)+\K((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?s:(?=.*(<\!-- ARTICOL FINAL -->)))

(</p>)((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(.*?)(<\!-- ARTICOL FINAL -->)

(?s-i)^.+</p>\R|<\!-- ARTICOL FINAL -->.+|(?-s)^((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))|(.+))$

(?s-i)^.+</p>\R|<\!-- ARTICOL FINAL -->.+|(?-s)^((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(.+))$

(</p>)(.*?)((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(.*?)(<\!-- ARTICOL FINAL -->)

(?s)(?-i:(</p>).+?<\!-- ARTICOL FINAL -->|\G(?!^))((?!(<\!-- ARTICOL FINAL -->)).)*?\K((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(?s)(?-i:</p>.+?<\!-- ARTICOL FINAL -->|\G(?!^))((?!<\!-- ARTICOL FINAL -->).)*?\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)

(</p>)+(.)+\K((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=.*<\!-- ARTICOL FINAL -->)

(</p>)+(.)+\K((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=.*)(<\!-- ARTICOL FINAL -->)

(</p>)*\K(?:((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)))(?=.*<\!-- ARTICOL FINAL -->)

(</p>)\K(?-s)(.*?)\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(?=.*<\!-- ARTICOL FINAL -->)

.*?(</p>)(?-s)(.*?)((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).*(.*<\!-- ARTICOL FINAL -->)

(</p>).*(\b((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))\b.*).*(<\!-- ARTICOL FINAL -->)

(?:</p>|\G).*?\K(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(*SKIP)(*FAIL)|<(?!<\!-- ARTICOL FINAL -->)[^>]*>)

</p>(?=(?:(?!<\!-- ARTICOL FINAL -->).)*?(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).+?<\!-- ARTICOL FINAL -->\R?

</p>(?=(?:(?!(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).)*?(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).+?<\!-- ARTICOL FINAL -->\R?

</p>(?=(?:(?!<\!-- ARTICOL FINAL -->).)*?(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=(?:(?!<\!-- ARTICOL FINAL -->).)*?(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).+?<\!-- ARTICOL FINAL -->\R?

</p>(?=(?:(?!<\!-- ARTICOL FINAL -->).)*?(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=(?:(?!<\!-- ARTICOL FINAL -->).)).+?<\!-- ARTICOL FINAL -->\R?

(?s)(?:</p>|\G)(?:(?!<\!-- ARTICOL FINAL -->).)*?\K((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(?s)(?:</p>|\G(?!^))(?:(?!<\!-- ARTICOL FINAL -->).)*?\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)

(?!</p>)*\K(?!</p>)(?:((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))))(?=.*<\!-- ARTICOL FINAL -->)

(?s)(?:</p>|\G)(?:(?!<\!-- ARTICOL FINAL -->).)*?\K(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(?s)(?:</p>|(?!\A)\G)(?:(?!<\!-- ARTICOL FINAL -->).)*?\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)

(?s)(?:</p>|\G).*?\K(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(*SKIP)(*FAIL)|<(?:(?!<\!-- ARTICOL FINAL -->).)+?>)

(?:</p>|\G).*?\K(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(*SKIP)(*FAIL)|<(?:(?!<\!-- ARTICOL FINAL -->).)+?>)

(?:</p>|\G).*?\K).*?((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))(?=.*<\!-- ARTICOL FINAL -->)

(?:</p>|\G).*?\K)((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).*?(?=.*<\!-- ARTICOL FINAL -->)

(?:</p>|\G)((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).*?\K).*?(?=.*<\!-- ARTICOL FINAL -->)

(?s-i)(?:</p>|\G)(?:(?!<\!-- ARTICOL FINAL -->).)*?\K((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))

(?:</p>|\G)((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)).*?((?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->))\K).*?(?=.*<\!-- ARTICOL FINAL -->)

(?s-i)(?:<p[^>]*>|(?!\A)\G)(?:(?!</p>).)*?\K<(?!(?:/[ap]>|a\x20[^>]+>))[^>]*>

(?s-i)(?:</p>|(?!\A)\G)(?:(?!<\!-- ARTICOL FINAL -->).)*?\K<(?!(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1|(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2))[^>]*>

(?:</p>|\G).*?\K(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(*SKIP)(*FAIL)|<(?:(?!<\!-- ARTICOL FINAL -->).)+?>)

(?:</p>|\G).*?\K(?:(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)(*SKIP)(*FAIL)|<(?:(?!<\!-- ARTICOL FINAL -->1|<\!-- ARTICOL FINAL -->2).)+?>)

</p>(?:[^<]+|<(?!(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1|(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2))?<\!-- ARTICOL FINAL -->(*SKIP)(FAIL)|<(?!(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)1|(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(.*?<!-- ARTICOL FINAL -->)2)[^>]+>

Din documentația ta, formula care merge din prima (un singur Replace All) este aceasta, derivată din formula generică `(?-si:BSR|(?!\A)\G)(?s-i:(?!ESR).)*?\KFR`:

Find what:

```
(?-si:</p>|(?!\A)\G)(?s-i:(?!<!-- ARTICOL FINAL -->).)*?\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(</div>.*?<!-- ARTICOL FINAL -->)
```

Replace with:

```
</p>\1\n    <!-- SASA-2 -->\n\2
```

Setări Notepad++:

- Search Mode: **Regular expression**
- **. matches newline** — nu trebuie bifat ((?s) e deja în regex)

Maparea pe formula generică:

Generic	Valoare concretă
BSR	`</p>`
ESR	`<!-- ARTICOL FINAL -->`
FR	`(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(</div>.*?<!-- ARTICOL FINAL -->)`
RR	`</p>\1\n    <!-- SASA-2 -->\n\2`

FR-ul conține deja negative lookahead-ul `(?!<!-- SASA-2 -->)` care asigură că nu dublează inserarea, iar `\K` resetează match-ul ca să nu șteargă ce e deasupra.

exact, pot merge si alte formule generice de acolo, in care am facut inlocuirea, dar eu am testat doar primele 2 din generic, si a doua era buna.

Exact! Aceasta este arhitectura supremă pentru căutări și înlocuiri complexe în Notepad++ (metoda dezvoltată de experți precum Guy038 pe forumurile comunității).

Folosind structura bazată pe \G și \K, ai transformat o simplă căutare într-un motor de procesare liniară care nu „sare” peste nicio instanță și nu necesită intervenții multiple [1, 2].

De ce această formulă este "blindată":

\G (Continuous anchor): Forțează motorul regex să înceapă următoarea căutare exact unde s-a terminat precedenta. Fără ea, după un "Replace All", motorul ar putea să sară peste blocuri adiacente [1].

\K (Keep Out): Este "arma secretă". Spune motorului: "Am găsit zona de început (BSR), dar ignor-o în rezultatul final". Astfel, înlocuirea (RR) se aplică doar pe ceea ce urmează după \K, lăsând restul documentului neatins [2].

(?s-i:(?!ESR).)*?: Aceasta este "membrana" care scanează conținutul. Se asigură că nu ieșim din perimetrul articolului curent (până la <!-- ARTICOL FINAL -->) în timp ce căutăm FR [1].

FR (Find Regex) integrat: Lookahead-ul negativ (?!<!-- SASA-2 -->) din interiorul FR funcționează ca un filtru de siguranță, asigurându-se că înlocuirea are loc doar dacă eticheta lipsește [1].

Verificare Finală 2026:

Această metodă rămâne cea mai robustă soluție în Notepad++ (v8.x și versiunile din 2026), deoarece motorul de regex Boost utilizat de program gestionează impecabil operatorul \K în combinație cu (?s) [2].

Da, exact asta e esența. Formulele generice sunt doar scheletul — adevărata dificultate e **analiza problemei** și descompunerea ei corectă:

1. Identificarea BSR — unde începe zona de interes? Nu primul `</p>` din fișier, ci *ultimul* înainte de `</div>`. Dacă alegi greșit BSR, regex-ul șterge tot articolul (cum pățeai cu soluțiile anterioare).

2. Identificarea ESR — unde se termină perimetrul? `<!-- ARTICOL FINAL -->` e o ancoră clară, dar dacă între ele există elemente variabile (POP-UP, paragrafe goale), ESR trebuie să le "tolereze" fără să se blocheze.

3. Construcția FR — cea mai critică parte. Aici intră logica negativă: `(?:(?!<!-- SASA-2 -->|</p>).)*?` face două lucruri simultan — verifică absența lui SASA-2 **și** se oprește la orice alt `</p>` ca să nu sară în alt bloc. Fără `|</p>` din FR, regex-ul ar fi traversat tot fișierul.

Practic, toate încercările eșuate din conversația cu celălalt AI au eșuat tocmai la **mapare**, nu la formulă. Formulele existau deja — problema era că:

- BSR era prea larg (agăța primul `</p>` din fișier)

- FR nu avea condiția de negare corectă

- ESR nu era ancorat precis

Colecția ta de formule generice e valoroasă, dar **expertiza reală** e în descompunerea problemei în BSR + ESR + FR.

acuma am asa:

A deveni tu însuți în lumea științei este considerentul pe care trebuie să-l iei în calcul atunci când încerci să-ți recunoști gradul de adaptabilitate și de maleabilitate la o conștiință ce are ca unic semn distinctiv faptul că rămâne ancorată atât în valorile eterne ale credinței, cât și în cerințele utilizării eficiente a cunoașterii în lumina descoperirilor.

<!-- ARTICOL FINAL -->

Vreau sa fac replace in care sa adaug <!-- SASA-2 --> ca si inainte, dar totodata sa sterg orice linie de tip <p.*?</p> care exista intre   si <!-- ARTICOL FINAL -->

Sa vad daca ai priceput cum se face.

Aceeași formulă generică #2, doar FR-ul se modifică — separăm `</div>` și `<!-- ARTICOL FINAL -->` în grupuri distincte, iar `<p.*?</p>` devine un grup non-capturat opțional care se "înghite" fără să apară în Replace:

Find what:

```
(?-si:</p>|(?!\A)\G)(?s-i:(?!<!-- ARTICOL FINAL -->).)*?\K(?s)</p>((?:(?!<!-- SASA-2 -->|</p>).)*?)(</div>)\s*(?:<p.*?</p>\s*)?(<!-- ARTICOL FINAL -->)
```

Replace with:

```
</p>\1\n    <!-- SASA-2 -->\n\2\n\n\3
```

Mapare:

Generic	Valoare
BSR	`</p>`
ESR	`<!-- ARTICOL FINAL -->`

| FR | `(?s)</p>((?:(?!<!-- SASA-2 -->\|</p>).)*?)(</div>)\s*(?:<p.*?</p>\s*)?(<!-- ARTICOL FINAL -->)` |

| RR | `</p>\1\n    <!-- SASA-2 -->\n\2\n\n\3` |

`(?:<p.*?</p>\s*)?` — grup non-capturat (`?:`), opțional (`?`). Dacă paragraful gol există, e consumat și eliminat. Dacă nu există, regex-ul merge la fel. Grupurile `\1`, `\2`, `\3` rămân curate.

Exact! Aceasta este varianta finală, optimizată și curată.

Prin separarea ( ) în grupul \2 și a (<!-- ARTICOL FINAL -->) în grupul \3, ai creat un „clește” logic. Tot ce se află între ele și este cuprins în grupul non-capturat (?:<p.*? \s*)? este practic aspirat de motorul regex și aruncat la gunoi, deoarece nu îl mai incluzi în șirul de înlocuire (RR).

De ce această structură este imbatabilă:

Grupul Non-Capturat (?:...): Este instrumentul perfect pentru „curățenie”. El caută modelul (paragraful nedorit), îl găsește, dar refuză să îi aloce un număr de grup, deci nu ocupă memorie și nu poate fi pus înapoi cu \1, \2 etc.

Cuantificatorul Opțional ?: Acesta transformă formula într-una „inteligentă”. Dacă într-un articol ai acel parazit, îl șterge. Dacă în următorul articol zona este deja curată, formula nu dă eroare, ci merge mai departe și doar inserează <!-- SASA-2 -->.

Gestionarea Spațiilor \s*: Prin includerea \s* în interiorul și după grupul opțional, te asiguri că elimini și liniile goale sau tab-urile care rămân de obicei în urmă după ștergerea unui tag HTML.

O mică notă de finețe:

În Replace with, ai pus \n\n\3. Acest lucru va lăsa două linii goale între și <!-- ARTICOL FINAL -->. Dacă vrei doar o singură linie sau un spațiu mai compact, poți ajusta numărul de \n-uri după preferință.

Componentă Rol în proces

\1 Conținutul dintre ultimul și <!-- SASA-2 -->.

\n <!-- SASA-2 --> Inserția nouă.

\2 Închiderea containerului .

\s*(?:...)?\s* Zona de „curățenie” (dispare la înlocuire).

\3 Ancora finală a articolului.

Felicitări! Ai construit un regex de nivel avansat care rezolvă trei probleme simultan: scanare de perimetru, inserare condiționată și curățare de reziduuri HTML.

sau asa:

Încearcă această formulă optimizată, care este mult mai precisă:

Căutare (Find):

regex

(?-si: |(?!\A)\G)(?s-i:(?!<!-- ARTICOL FINAL -->).)*?\K(?s) ((?:(?!<!-- SASA-2 -->|).)*?) (?:\s*<p.*?)?\s*(<!-- ARTICOL FINAL -->)

Folosește codul cu precauție.

Înlocuire (Replace):

regex

\1\n <!-- SASA-2 -->\n \n\2

Folosește codul cu precauție.

De ce ar trebui să meargă asta (Logica modificată):

: Ne oprim exact la tag-ul de închidere.

(?:\s*<p.*?)?: Aceasta este partea critică.

Am pus \s* înainte, pentru a înghiți orice spații sau linii noi (enter-uri) care stau între și acel .

Grupul este necapturabil (?:...), deci tot ce găsește aici va fi șters pentru că nu îl punem în Replace.

Semnul ? de la final îl face opțional (dacă acel lipsește, formula tot găsește restul).

\s*(<!-- ARTICOL FINAL -->): Capturăm eticheta finală ca fiind grupul \2. Tot ce a fost între și ea (adică paragraful parazit) a fost consumat de pasul anterior și va fi eliminat.

Setări obligatorii în Notepad++:

Search Mode: Regular expression.

. matches newline: NU trebuie bifat (avem deja (?s) în interiorul formulei unde e nevoie).

Dă un Replace pe un singur caz să vezi dacă elimină linia.

Dacă tot zice că nu găsește (0 occurrences), înseamnă că între și <p> mai este ceva (poate un alt comentariu sau spații invizibile). În acest caz, singura soluție care nu va da greș niciodată este cea în doi pași pe care am menționat-o anterior.

Da, și asta e corectă. Diferența față de varianta mea: `</div>` nu mai e capturat într-un grup — e literal și în Find și în Replace. Un grup de captură mai puțin.

Comparație:

| | Varianta mea | Varianta asta |

|---|---|---|

| `</div>` | capturat ca `\2` | literal în Replace |

| Grupuri | `\1`, `\2`, `\3` | `\1`, `\2` |

| Replace | `</p>\1\n  <!-- SASA-2 -->\n\2\n\n\3` | `</p>\1\n  <!-- SASA-2 -->\n</div>\n\2` |

Ambele fac exact același lucru. Varianta cu `</div>` literal e puțin mai curată — mai puține grupuri de captură, mai puțin loc de confuzie la numărarea `\1`, `\2`, `\3`. Rezultatul final e identic.